Remarks

In the Office Action dated 12/02/2006 ("Office Action"), Claims 1-32 were rejected. In the Amendment set forth above, Claims 1, 2, 16, 17, 25, 28, 30 are amended, and the remaining Claims are as originally presented. In view of the amendments to the Claims and the remarks set forth below, it is respectfully submitted that all Claims are in condition for Allowance.

1. Claims 1-14, 16-26 and 28-37 were rejected under 35 USC §102(b) as being anticipated by U.S. Patent No. 5,913,224 to MacDonald ("MacDonald"). This rejection is respectfully traversed.

Before discussing the specific Claim language, Applicants' invention is summarized for discussion purposes. Applicants' invention relates to a memory system and method. In one embodiment of the invention, data obtained from a main memory may be stored within a processor node 120 that includes one or more instruction processors (IPs) and one or more caches. (See, for example, Applicants' Figure 1.) For any given data item retained by the processor node, cache tag logic associated with the processor node may, but need not, record the presence of that data item. Whether the cache tag logic records the presence of any given data item depends on the state of one or more programmable indicators. These indicators may be used to identify various attributes associated with the data item. Such attributes may include the type of operation that caused the data to be provided to the processor node (and hence the cache tag logic), either from the main memory or from one of the IPs. As another example, attributes may relate to which requester (e.g., IP) requested the data from main memory. Additional attributes are described in the Specification. Before continuing, it may be noted that this summary is intended to be entirely consistent with the information described in the Specification and Drawings.

Turning now to Claim 1, this amended Claim appears as follows:

> A memory system, comprising:
> a programmable storage device to store one or more indicators;
> a cache;
> cache tag logic; and
> a control circuit coupled to the storage device, the cache, and to the cache tag logic, the control circuit to receive data for possible retention in the cache and to determine, based on the state of the one or more indicators, whether to update the cache tag logic to track the data.

As described in Claim 1, the control circuit receives data. Based on the state of one or more indicators stored in the programmable device, the control circuit will determine whether the cache tag logic will be updated to track this data.

At most, MacDonald teaches use of programmable lock bits that control *where* in the MacDonald cache newly-retrieved data may be stored. For example, in MacDonald, data retrieved from system memory will not be stored within locked cache lines. However, in MacDonald, there are no programmable indicators that determine *whether* data should be stored within the cache, or whether tag logic should be updated to track data, as is described in Claim 1. In other words, when a cache miss occurs in MacDonald such that data is retrieved from system memory, the data will be stored to cache. The MacDonald lock bits control where in cache this data will be stored, but not whether it will be so stored.

Next, the Examiner's assertions regarding Claim 1 are discussed. First, the Examiner cites the MacDonald cache tag logic of column 8 line 31 as teaching Applicants' cache tag logic, and cites the MacDonald cache management unit of column 5 lines 47-48 as teaching Applicants' control circuit. (Office Action page 2, section 2, lines 7-8.) The cited MacDonald cache tag logic is included within the MacDonald embodiment of Figure 5, whereas the cited MacDonald cache management unit is included in the entirely different embodiment of Figure 2. There is no indication in MacDonald that the MacDonald cache tag logic of the embodiment of Figure 5 is used in the

embodiment of Figure 2, and vice versa. In particular, the cache management unit of Figure 2 does not determine whether to update the cache tag logic of Figure 5 based on any programmable indicators. Thus, the aspects cited by the Examiner do not teach Applicants' invention of Claim 1, and for at least this reason, the rejection is improper.

Next, it may be noted that the MacDonald description does mention that the MacDonald cache management unit 202 of MacDonald Figure 2 includes address tags, even though these tags are not shown, and are not described in any detail. Some consideration is therefore given as to whether those briefly-mentioned address tags teach Applicants' cache tag logic and the cache management unit 202 teaches Applicants' control circuit. In this regard, nothing in MacDonald describes an aspect wherein the MacDonald cache management unit (said to be akin to Applicants' control circuit) uses programmable indicators to determine whether to update the MacDonald address tags.

The MacDonald description appears to indicate that the MacDonald cache operates as would a typical set associative cache configuration which updates the tags every time data is provided to the cache management unit 202 for storage within the cache. (See, for example, MacDonald column 5 lines 59-63.) This is specifically discussed in regards to the MacDonald embodiment of Figure 5 which describes that whenever a cache miss occurs, the data is provided to the cache and the tag logic is then updated to include the tag associated with the new data. (MacDonald column 10 lines 15-17.)

To summarize, nothing in MacDonald discusses any operation whereby the cache tags, may, or may not, be updated to track data that has been presented to the cache management unit (and hence the cache). For at least this reason, Claim 1 is allowable both as previously presented, and as currently presented.

Next, the Examiner's assertions regarding Applicants' programmable indicators of Claim 1 are considered. The Examiner states that these indicators are taught by the title of the application, which declares that the invention is a programmable cache. (Office action, section 2, lines 5-6.) In regards to this

statement, it is noted that the programmable nature of the cache involves the lock bits. These bits prevent cache lines that contain real-time code from being replaced according to a least-recently used algorithm, and having nothing to do with whether cache tags are updated. This is described as follows:

> "...the cache subsystem locks the code into the cache preventing overwriting the code with more recently used data. The real-time code is locked into cache by setting a lock bit associated with each line of cache containing the real-time code." (MacDonald column 3 lines 7-11.)

This passage reiterates that the lock bits determine *where* newly-retrieved memory data is stored within the cache. That is, newly-retrieved data will always be stored to unlocked, versus locked, cache lines. Stated otherwise, nothing in MacDonald indicates that locking some cache lines will ever prevent newly-retrieved data or code from being stored within cache. This newly-retrieved information is always stored to the cache in unlocked locations. Thus, there are no MacDonald indicators that are used to determine *whether* to store data within the MacDonald cache, or *whether* to update the MacDonald tags to track data.

A similar analysis can be performed in regards to the programmable device that is included in the second MacDonald embodiment shown in Figure 5. This real-time address register 265 controls which cache locations contain real-time code. (MacDonald column 9 lines 30-35.) Thus, this register may control where to store newly-retrieved code or data, but not whether to store the code or data. This device does not teach Applicants' programmable indicators used to determine whether to update cache tags to track data that has been provided to the cache, as claimed by Applicants' Claim 1.

Next, the Examiner's assertions concerning Applicants' control circuit and programmable indicators are considered further. The Examiner states Applicants' control circuit is taught by MacDonald column 5 lines 3-8, since

> "... the cache management unit directs data transfers in and out of the L2 cache, and orchestrates the transfer of data, address and control signals between local bus and system memory, and also *includes a memory controller (a.k.a. indicator)*)" (Office Action page 2, last sentence, emphasis added.)

It appears from the foregoing statement that the Examiner is citing the MacDonald memory controller as teaching Applicants' programmable indicators. The MacDonald memory controller is included in the MacDonald cache management unit 202, and provides access to the L2 cache memory 201. This memory controller may be any one of the commonly known memory controllers compatible with the selected CPU core and may be located as part of the processor. (MacDonald column 5 lines 6-13.) There is no indication in MacDonald that anything in this memory controller is programmable to control whether the cache tags are updated. Therefore, it is not understood how this memory controller could teach Applicants' programmable indicators used to determine whether the cache tags are updated to track data. If this rejection is maintained, further clarification is requested regarding the significance of the MacDonald memory controller in regards to Applicants' indicators.

For all of the foregoing reasons, nothing in MacDonald teaches Applicants' invention of Claim 1, both as previously, and as currently, presented. This rejection is improper, and should be withdrawn. If this rejection is not withdrawn, more explanation is respectfully requested regarding the programmable aspect of MacDonald that is thought to be relevant to determining whether cache tags are updated to track data, as claimed by Claim 1.

Claims 2-14 depend from Claim 1, and are allowable for at least the reasons set forth in regards to Claim 1. These Claims include other aspects not taught by MacDonald as follows:

Claim 2 describes circuits to determine, based on the one or more programmable indicators, whether to store the data to the cache. As discussed above, it appears in MacDonald that wherever data is provided to the MacDonald cache, a replacement operation occurs. There is no indication in MacDonald that such an operation occurs based on the state of any programmable indicators.

The Examiner states that this aspect of the invention is taught by MacDonald column 5 lines 54-57. (Office Action page 3.) This passage indicates that the MacDonald cache management unit "may contain additional conventional circuits to control well-known caching functions such as read, write,

update, invalidation, copy-back, and flush operations." However, nothing in this passage or MacDonald generally describes determining whether to store the data to cache based on programmable indicators. As previously discussed, at most the MacDonald lock bits determine *where* in the cache to store data that has been newly-retrieved from memory, but these bits do not determine *whether* to store the data to cache, as is claimed by Claim 2. Moreover, the type of operation described in Claim 2 is not a "well-known caching function". For this additional reason, Claim 2 is allowable over the current rejection.

Claim 3 includes the aspect whereby the indicators describe the cache as being unavailable for use. The Examiner states that this aspect of the invention is taught by MacDonald column 4 lines 60-67. (Office Action page 3.) The cited passage describes the TLB real-time lock bits used by the processor to indicate to the cache when real-time code is being retrieved.

Neither the MacDonald TLB lock bits, nor the lock bits within cache, indicate that the cache is unavailable for use. The MacDonald lock bits are used to ensure that certain cache locations are used solely for the real-time code. Stated otherwise, locked cache lines ***are available for use***, although they are exclusively used to store the real-time code that is being executed by the CPU at that time. Nothing in MacDonald teaches or suggests indicators to indicate the cache is not available for use, and for this additional reason, Claim 3 is allowable over this rejection, which is improper, and should be withdrawn.

Claim 4 describes a main memory to provide to the cache requested data that is not stored within the cache, and a circuit that may replace the data in the cache based on the state of the indicators. The Examiner states that this aspect of the invention is taught by MacDonald column 5 lines 47-52. (Office Action page 3.) The cited passage of MacDonald describes the snoop write-back circuit that controls the writing of dirty data *from the L2 cache to the memory* based on whether it has been updated. Such write-back operations are well-known in the art. For example, see U.S. Patent No. 5,802,559 assigned to Advanced Micro

Devices, the assignee of MacDonald. The '559 patent describes a system that appears related to that described in MacDonald. In the '559 patent, the snoop write-back circuit is described in detail as a circuit that writes dirty data *from* the cache back to the system *memory*. (As an example, see Claim 1 of the '559 patent which describes the snoop write-back circuit as a circuit configured "to control a write-back to said system memory of dirty data stored within said cache memory".)

As may be appreciated from the foregoing, the snoop write-back circuit briefly mentioned in MacDonald has nothing to do with providing data *from memory to cache*, but rather has to do with storing data from cache to memory. Further, this circuit has nothing to do with determining whether to store data to a cache based on programmable indicators. For these additional reasons, Claim 4 is allowable over this rejection.

Claims 5 and 6 involve indicators that identify one or more of the request types, and a control circuit that prevents the replacement of the data in the cache if the data was provided in response to any of the identified request types. These aspects are said to be taught by MacDonald column 5 lines 47-58. (Office Action page 3, last line.) The cited passage mentions in a very cursory manner "well-known caching functions such as read, write, updates,..." etc., but does not describe that replacement of data in the cache is based on identified request types as identified by programmable indicators. If this rejection is maintained, more clarification is respectfully requested regarding the perceived significance of the cited MacDonald passage.

Further in regards to the request types, the Examiner cites the aspect of MacDonald wherein virtual addresses are translated to physical addresses. (Office Action page 4 in regards to Claim 6.) Again, this does not appear to have anything to do with one or more request types used to determine whether data is stored to cache. More clarification is requested regarding the significance of this address translation function.

MacDonald does not teach the additional aspects of Claims 5 and 6, which are allowable as presently presented for this additional reason.

Claim 7 relates to indicators that identify one or more requesters. The control circuit replaces the data in the cache if the data was returned from the main memory in response to a request issued by any of the identified requesters. The Examiner cites column 5 lines 47-52 as teaching this aspect. This passage describes a write-back circuit that writes dirty data from the L2 cache back to memory, as is known in the art. As discussed above, this has nothing to do with retrieving data from memory for storage in cache. The additional aspects of Claim 7 are not taught by MacDonald, and for this additional reason, this Claim is allowable.

Claim 8 involves the main memory providing data to the cache with a response that is any one of multiple response types. At least one of the indicators identifies one or more of the response types, and the control circuit replaces the data in the cache if the data is returned from the main memory with any of the identified response types. The Examiner states that this aspect is taught by MacDonald column 2 lines 38-46. (Office Action page 4.) The cited passage describes that in response to a request for data, the data is replaced in cache. It is not understood how this relates in any way to the memory providing data with a response that is any one of multiple type, and wherein the replacement occurs based on response types identified by the programmable indicators. If this rejection is maintained, more clarification as to the perceived significance of the cited passage is respectfully requested. This Claim is allowable over this rejection for the additional aspects cited by Claim 8.

Claim 9 depends from Claim 2 and is allowable for at least the reasons discussed in regards to Claims 1 and 2 set forth above.

Before continuing, it is noted that Claim 9 further describes at least one requester that is coupled to the control circuit to return data to the cache tag

logic. The Examiner appears to cite column 10 lines 9-17 of MacDonald related to the memory interface as teaching this aspect. This memory interface receives data from the system memory 300. Therefore, the Examiner appears to be citing the MacDonald memory interface (and possibly MacDonald system memory 300) as teaching Applicants' requester of Claim 9. This is discussed further in regards to Claim 12 below.

Turning next to Claims 10 and 11, these Claims describe that the requester of Claim 9 returns data to the cache tag logic during an operation that is any one of multiple operation types. The control circuit stores the data to the cache if the data is returned during any of the operation types identified by the indicators. The Examiner cites column 10 lines 44-55 of MacDonald as teaching this aspect of the invention. The cited MacDonald passage relates to the situation wherein a cache miss occur, and the data being retrieved from memory is not real-time code. Therefore it is replaced in data-way 0. Again, this type of operation controls *where* to store the data (in this case data way 0), and *not whether* to store the data to cache, as claimed by Applicants' Claim 10 and 11. For this additional reason, Claims 10 and 11 are allowable over the current rejection.

Claim 12 relates to a main memory coupled to the control circuit. The control circuit is adapted to forward data returned by the requester to the main memory based, at least in part, on the state of at least one of the indicators. The Examiner cites MacDonald column 8 lines 6-8 as teaching this aspect of the invention. The cited passage relates to page allocation and TLB entries, which are updated if the code copied into the system memory is real-time code.

The cited passage relates to how the TLB entries are initialized by the operating system when real-time code is copied into main memory such as at system initialization time. Generally, code would be copied from mass storage (e.g., disks) at this time. This passage has nothing to do with copying code from any requester to main memory, as described by Applicants' Claim 12.

Moreover, recall that the Examiner appeared, in regards to Claims 9, to assert that the MacDonald memory interface 295 teaches Applicants' requester. This assertion appears entirely inconsistent with the passage cited in relation to Claim 12. Finally, the passage cited in relation to Claim 12 has nothing to do with performing the copy operation from the requester to memory based in part of the programmable indicators. For at least these reasons, the additional aspects of Claim 12 are not taught by MacDonald, and this Claim is allowable for these additional reasons.

Claim 13 describes the aspect wherein memory coherency actions may be incomplete for the returned data or for associated data retained by the at least one requester or the cache. A request tracking circuit is coupled to the control circuit to prevent the returned data from being forwarded to the main memory until all of the memory coherency actions have been completed for the returned data or for the associated data. The Examiner cites column 6 lines 24-42 as teaching this aspect of the invention. The cited passage describes the four steps that facilitate execution of real-time code. Namely, the processor first directs the block of code to be stored in cache. The cache then locks the real-time code to avoid overwriting. The processor executes the code, and the cache then unlocks the code to free up the cache locations that store the code.

The above-described passage has nothing to do with memory coherency actions performed by main memory. Moreover, it is not understood how this passage teaches a request tracking circuit to prevent the return of data to main memory before all memory coherency actions has been completed. Nothing in MacDonald teaches the additional aspects of Claim 13, and this Claim is allowable for this additional reason.

Claim 14 describes that the programmable storage device further stores microcode that is used to control the control circuit. The Examiner cites column 6 lines 29-34 as teaching this aspect. This passage describes that the cache stores the real-time code that is executed by the CPU. Recall that the Examiner cites the cache management unit as teaching applicants' control circuit. (Office

Action page 2 line 9.) However, the code stored within the cache is being executed (that is, this code is controlling) the MacDonald processor, and is not controlling the MacDonald cache management unit, as follows:

> "...the processor 101 executes the real-time code after it has been stored in L2 cache memory 201." (MacDonald column 6 lines 36-38.)

Therefore, it appears that the rejections of Claim 1 and 14 are not consistent. In the rejection of Claim 1, the Examiner is stating that the cache management unit teaches Applicants' control circuit, whereas in the rejection of Claim 14, the Examiner implies that the processor teaches the control circuit. At any rate, neither of these MacDonald elements teaches a circuit for updating cache tags based on programmable indicators. Because the additional aspects of Claim 14 are not taught by MacDonald, Claim 14 is allowable for these additional reasons.

Turning next to independent Claim 16, this Claim has been amended to clarify the aspects of the invention. For reasons similar to those discussed above in regards to Claim 1, this Claim is not taught by MacDonald. This rejection is improper, and should be withdrawn.

Claims 17-26 depend from Claim 16, and are allowable over this rejection for at least the reasons discussed above in regards to Claims 1 and 16. These Claims include additional aspects similar to those discussed above in regards to Claims 2-14. For additional reasons similar to those discussed above in regards to Claims 2-14, these Claims are allowable over this rejection.

Next, independent Claim 28 is considered. This Claim includes aspects similar to those discussed in regards to Claim 1. For reasons similar to those discussed above in regards to Claim 1, this Claim is not taught by MacDonald. This rejection is improper, and should be withdrawn.

Claims 29-37 depend from Claim 28, and are allowable over this rejection for at least the reasons discussed above in regards to Claims 1 and 28. These Claims include additional aspects similar to those discussed above in regards to

Claims 2-14. For additional reasons similar to those discussed above in regards to Claims 2-14, these Claims are allowable over this rejection.

2. Claims 15 and 27 were rejected under 35 USC §103(a) as being unpatentable over MacDonald as applied to Claims 1 and 16 above. This rejection is respectfully traversed.

Claims 15 and 27 depend from independent Claims 1 and 16 respectively, and are allowable over this rejection for at least the reasons set forth above in regards to Claim 1. Moreover, these Claims describe the aspect whereby at least one of the indicators is automatically re-programmed based on monitored system conditions.

The Examiner states that MacDonald describes a write-back circuit that replaces data within the L2 cache based on a certain state, and therefore stating that this is done automatically does not change the purpose of the functionality of the claimed invention.

The Examiner's rejection of Claims 15 and 27 is improper for several reasons. First, as previously discussed, it is well-known that a snoop write-back circuit writes dirty data from cache back to main memory. (See, for example, other patents assigned to AMD such as the '559 patent discussed above.) Moreover, the MacDonald write-back circuit does not monitor conditions within system memory and re-program indicators (either automatically or otherwise) based on those monitored conditions. Finally, the MacDonald write-back circuit has nothing to do with any indicators that are used to determine whether to update tags to track data for a cache. Neither the MacDonald write-back circuit, nor anything else in MacDonald, teaches or even suggests this functionality. Therefore Claims 15 and 27 are allowable over this rejection for these additional reasons.

Conclusion

In the Office Action dated 12/02/2006, Claims 1-32 were rejected. In the Amendment set forth above, Claims 1, 2, 16, 17, 25, 28, 30 are amended to clarify aspects of the invention, and the remaining Claims are as originally presented. In view of the amendments to the Claims and the remarks set forth above, it is respectfully submitted that all Claims are in condition for Allowance, and a Notice of Allowance is respectfully requested. If the Examiner has any questions regarding the subject Application or this response, a call to the undersigned is encouraged and welcomed.

Respectfully submitted,

Beth L. McMahon

Beth L. McMahon
Attorney for Applicants
Reg. No. 41,987
Telephone No. (651) 635-7893

Unisys Corporation
M.S. 4773
P.O. Box 64942
St. Paul, MN 55164-0942